On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks *

Sung-Ju Lee ** William Su *** Mario Gerla

Computer Science Department, University of California, Los Angeles, CA 90095-1596, USA
E-mail: (sjlee, wsu, gerla)@cs.ucla.edu

An ad hoc network is a dynamically reconfigurable wireless network with no fixed infrastructure or central administration. Each host is mobile and must act as a router. Routing and multicasting protocols in ad hoc networks are faced with the challenge of delivering data to destinations through multihop routes in the presence of node movements and topology changes. This paper presents the On-Demand Multicast Routing Protocol (ODMRP) for wireless mobile ad hoc networks. ODMRP is a mesh-based, rather than a conventional tree-based, multicast scheme and uses a forwarding group concept; only a subset of nodes forwards the multicast packets via scoped flooding. It applies on-demand procedures to dynamically build routes and maintain multicast group membership. ODMRP is well suited for ad hoc wireless networks with mobile hosts where bandwidth is limited, topology changes frequently, and power is constrained. We evaluate ODMRP performance with other multicast protocols proposed for ad hoc networks via extensive and detailed simulation.

Keywords: Multicast, routing, ad hoc networks, mobile computing

1. Introduction

Multipoint communications [13] have emerged as one of the most researched areas in the field of networking. As the technology and popularity of Internet grow, applications, such as video conferencing, that require multicast support are becoming more widespread. Another interesting recent development has been the emergence of dynamically reconfigurable wireless ad hoc networks [19,21] to interconnect mobile users. Ad hoc networks have no fixed infrastructure or central administration, and each host must communicate one another via packet radios. Due to the limited radio propagation range of wireless devices, routes are often "multihop." Applications such as disaster recovery, crowd control, search and rescue, and automated battlefields are typical examples of where ad hoc networks are deployed. Nodes in these networks move arbitrarily, thus making the network topology to change frequently and unpredictably. Moreover, bandwidth and battery power are limited. Hence, efficient utilization of routing packets and immediate recovery of route breaks are critical in routing and multicasting protocols for ad hoc networks.

In a typical ad hoc environment, network hosts work in groups to carry out a given task. Therefore, multicast plays an important role in ad hoc networks. Multicast protocols used in static networks (e.g., Distance Vector Multicast Routing Protocol (DVMRP) [11], Multicast Open Shortest Path First (MOSPF) [31], Core Based Trees (CBT) [4], and Protocol Independent Multicast (PIM) [12]) do not perform well in wireless ad hoc networks because multicast tree structures are fragile and must be readjusted as connectivity changes. Furthermore, multicast trees usually require a global routing substructure such as link state or distance vector. The frequent exchange of routing vectors or link state tables, triggered by continuous topology changes, yields excessive channel and processing overhead.

To overcome these limitations, the On-Demand Multicast Routing Protocol (ODMRP) [25] has been developed. ODMRP is a mesh-based, instead of a tree-based, multicast protocol that provides richer connectivity among multicast members. By building a mesh

* This work was funded in part by the Defense Advanced Research Projects Agency (DARPA) under contract DAAB07-97-C-D321, as a part of the Global Mobile Information Systems (GloMo) program.

** Now with Hewlett-Packard Laboratories, Palo Alto, CA.
*** Now with eWings Technologies, Plano, TX.

and supplying multiple routes, multicast packets can be delivered to destinations in the face of node movements and topology changes. In addition, the drawbacks of multicast trees in mobile wireless networks (e.g., intermittent connectivity, traffic concentration, frequent tree reconfiguration, non-shortest path in a shared tree, etc.) are avoided. To establish a mesh for each multicast group, ODMRP uses the concept of forwarding group [7]. The forwarding group is a set of nodes responsible for forwarding multicast data on shortest paths between any member pairs. ODMRP also applies on-demand [9,29] routing techniques to avoid channel overhead and improve scalability. A soft-state approach is taken to maintain multicast group members. No explicit control message is required to leave the group. We believe the reduction of channel/storage overhead and the richer connectivity make ODMRP more attractive in mobile wireless networks.

A few other multicast routing protocols have been recently proposed for ad hoc networks [6–8,14,17,20,34,38,39,42]. The Reservation-Based Multicast (RBM) routing protocol [8] builds a core (or a Rendezvous Point) based tree for each multicast group. RBM is a combination of multicast, resource reservation, and admission control protocol where users specify requirements and constraints. The Lightweight Adaptive Multicast (LAM) algorithm [20] is a group shared tree protocol that does not require timer-based messaging. Similar to other core-based protocols, it suffers from disadvantages of traffic concentration and vulnerability of the core. The Adhoc Multicast Routing Protocol (AMRoute) [6] is also a shared-tree protocol which allows dynamic core migration based on group membership and network configuration. The Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) [42] builds a shared-tree to deliver multicast data. Each node in the multicast session is assigned an ID number and it adapts to connectivity changes by utilizing the ID numbers. A multicast extension of Ad Hoc On Demand Distance Vector (AODV) routing protocol has been proposed in [38]. Its uniqueness stems from the use of a destination sequence number for each multicast entry. The sequence number is generated by the multicast grouphead to prevent loops and to discard stale routes. Similar to ODMRP, the Core-Assisted Mesh Protocol (CAMP) [14] uses a mesh. However, a conventional routing infrastructure based on enhanced distance vector algorithm (e.g., Wireless Routing Protocol (WRP) [32]) or link state algorithm (e.g., Adaptive Link-State Protocol (ALP) [16] or Source Tree Adaptive Routing (STAR) [15]) is required for CAMP to operate. Core nodes are used to limit the traffic required when a node joins a multicast group.

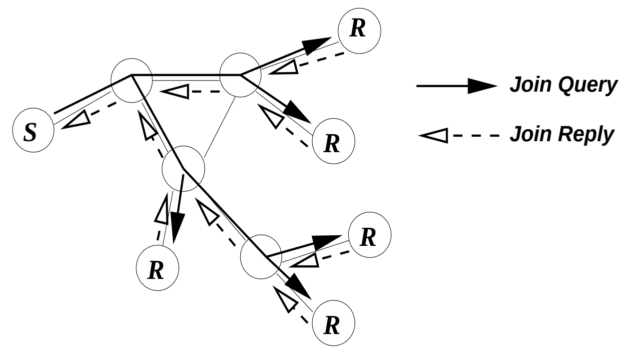


Figure 1. On-demand procedure for membership setup and maintenance.

The remainder of the paper is organized as follows. Section 2 illustrates the protocol in detail. Protocol performance improvements using mobility prediction are introduced in Section 3. Then, by extensive simulation, we evaluate and compare the performance of ODMRP with some of the above mentioned ad hoc multicast protocols. Section 4 describes the simulation model and methodology followed by simulation results and analysis in Section 5. Concluding remarks are made in Section 6.

2. On-Demand Multicast Routing Protocol

2.1. Multicast Route and Mesh Creation

In ODMRP, group membership and multicast routes are established and updated by the source "on demand." Similar to on-demand unicast routing protocols, a request phase and a reply phase comprise the protocol (see Figure 1). While a multicast source has packets to send, it floods a member advertising packet with data payload piggybacked. This packet, called JOIN QUERY, is periodically broadcasted to the entire network to refresh the membership information and update the routes as follows. When a node receives a non-duplicate JOIN QUERY, it stores the upstream node ID (i.e., backward learning) into the routing table and rebroadcasts the packet. When the JOIN QUERY packet reaches a multicast receiver, the receiver creates and

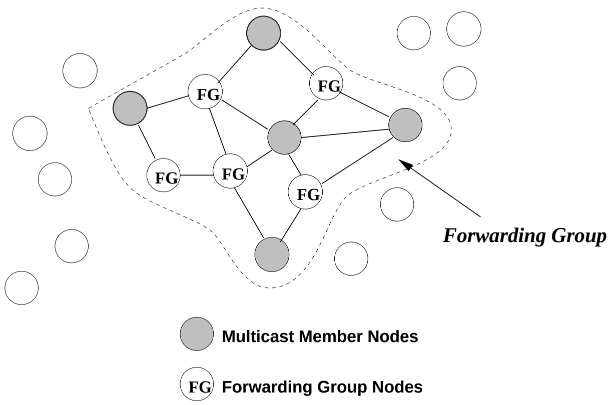


Figure 2. The forwarding group concept.

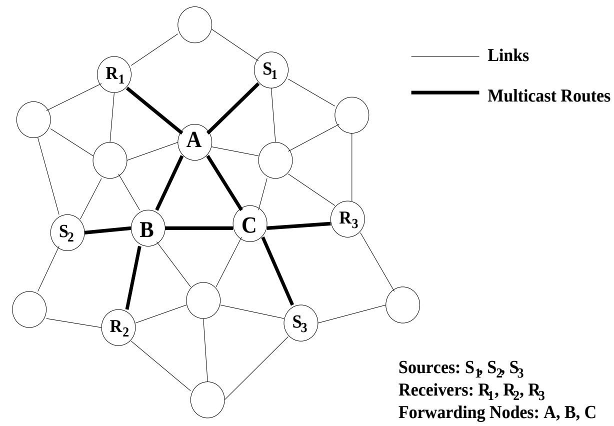


Figure 3. Why a mesh?

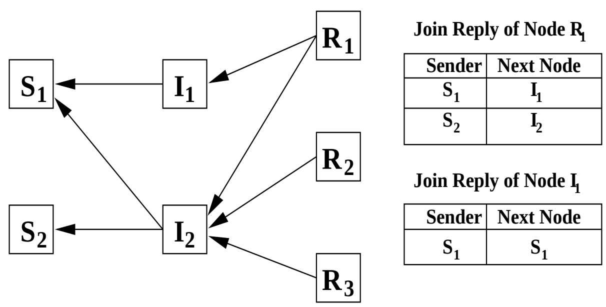


Figure 4. An example of a JOIN REPLY forwarding.

broadcasts a JOIN REPLY to its neighbors. When a node receives a JOIN REPLY, it checks if the next node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets the `FG_FLAG` (Forwarding Group Flag) and broadcasts its own JOIN REPLY built upon matched entries. The JOIN REPLY is thus propagated by each forwarding group member until it reaches the multicast source via the shortest path. This process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the "forwarding group."

We have visualized the forwarding group concept in Figure 2. The forwarding group is a set of nodes which is in charge of forwarding multicast packets. It supports shortest paths between any member pairs. All nodes inside the "bubble" (multicast members and forwarding group nodes) forward multicast data packets. Note that a multicast receiver also can be a forwarding group node if it is on the path between a multicast source and another receiver. The mesh provides richer connectivity among multicast members compared with trees. Route redundancy among forwarding group helps overcome node displacements and channel fading. Hence, unlike trees, frequent reconfigurations are not required.

Figure 3 is an example to show the robustness of a mesh configuration. Three sources (S_1, S_2, and S_3) send multicast data packets to three receivers (R_1, R_2, and R_3) via three forwarding group nodes (A, B, and C). Suppose the route from S_1 to R_2 is $<S_1$-A-B-$R_2>$. In a tree configuration, if the link between nodes A and B breaks or fails, R_2 cannot receive any packets from S_1 until the tree is reconfigured. ODMRP, on the other hand, already has a redundant route in $<S_1$-A-C-B-$R_2>$ to deliver packets without going through the broken link between nodes A and B.

2.2. Example

Let us consider Figure 4 as an example of a JOIN REPLY forwarding process. Nodes S_1 and S_2 are multicast sources, and nodes R_1, R_2, and R_3 are multicast receivers. Nodes R_2 and R_3 send their JOIN REPLIES to both S_1 and S_2 via I_2. R_1 sends its JOIN REPLY to S_1 via I_1 and to S_2 via I_2. When receivers send their JOIN REPLIES to next hop nodes, an intermediate node I_1 sets the `FG_FLAG` and builds its own JOIN REPLY since there is a next node ID entry in the JOIN REPLY received from R_1 that matches its ID. Note that the JOIN REPLY built by I_1 has an entry for sender S_1 but not for S_2 because the next node ID for S_2 in the received JOIN REPLY is not I_1. In the meantime, node I_2 sets the `FG_FLAG`, constructs its own JOIN REPLY and sends it to its neighbors. Note that even though I_2 receives three JOIN REPLIES from the receivers, it broadcasts the JOIN REPLY only once because the second and third table arrivals carry no new source information. Channel overhead is thus reduced dramatically in cases where numerous multicast receivers share the same links to the source.

2.3. Reliability

The reliable transmission of JOIN REPLIES plays an important role in establishing and refreshing multicast routes and forwarding groups. Hence, if JOIN REPLIES are not properly delivered, effective multicast routing cannot be achieved by ODMRP. The IEEE 802.11 MAC (Medium Access Control) protocol [18], which is the emerging standard in wireless networks, performs reliable transmission by retransmitting the packet if no acknowledgment is received. However, if the packet is broadcasted, no acknowledgments or retransmissions are sent. In ODMRP, the transmissions of JOIN REPLY are often broadcasted to more than one upstream neighbors since we are handling multiple sources (e.g., see the JOIN REPLY from node R_1 in Figure 4). In such cases, the hop-by-hop verification of JOIN REPLY delivery and the retransmission cannot be handled by the MAC layer. It must be done indirectly by ODMRP. Another option for reliable delivery is to subdivide the JOIN REPLY into separate sub-tables, one for each distinct next node. In Figure 4 for example, the JOIN REPLY at node R_1 is split into two JOIN REPLIES, one for neighbor I_1 and one for neighbor I_2. These JOIN REPLIES are separately unicasted using a reliable MAC protocol such as IEEE 802.11 or MACAW [5]. Since the number of neighbors is generally limited (typically, about six neighbors is the optimum in a multihop network [23]), the scheme still scales well to large number of sources. This option can actually be used as a backup to the passive acknowledgment option as discussed below.

We adopt a scheme that was used in [21]. Figure 5 is shown to illustrate the mechanism. When node B transmits a packet to node C after receiving a packet from node A, node A can hear the transmission of node B if it is within B's radio propagation range. Hence, the packet transmission by node B to node C is used as a "passive acknowledgment" to node A. We can utilize this passive acknowledgment to verify the delivery of a JOIN REPLY. Note that the source itself must send an active acknowledgment to the previous hop since it does not have any next hop to send a JOIN REPLY to unless it is also a forwarding group node for other sources.

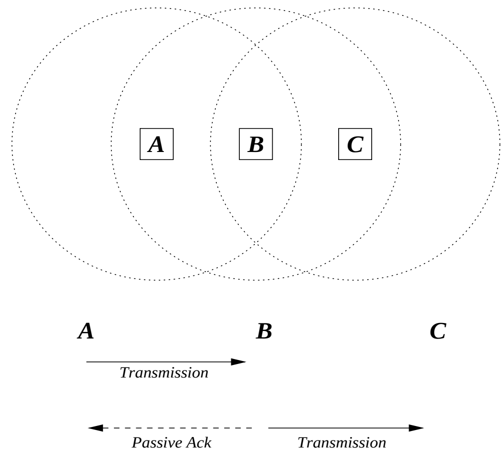


Figure 5. Passive acknowledgments.

Considering the case in Figure 4 again, we note that once the nodes I_1 and I_2 receive the JOIN REPLY from node R_1, they will construct and forward their own JOIN REPLY to next hops (in this case, sources S_1 and S_2). In transmitting their JOIN REPLY, nodes I_1 and I_2 may overlap with each other. If I_1 and I_2 are within receiving range, they will recover because of the carrier sense feature in CSMA (Carrier Sense Multiple Access) [24]. However, if they are out of range, they will be unaware of the "hidden terminal" condition of node R_1, which cannot hear the (overlapped) passive acknowledgments. Thus, a node may not hear the passive acknowledgments of its upstream neighbor because of conflicts due to the hidden terminal problem. It will also not hear the passive acknowledgment if the upstream neighbor has moved away. In either case, when no acknowledgment is received within the timeout interval, the node retransmits the message. Note that the node may get acknowledgments from some, but not all upstream neighbors. As an option, the retransmission could be carried out in unicast mode, to selected neighbors, with reduced sub-tables. If packet delivery cannot be verified after an appropriate number of retransmissions, the node considers the route to be invalidated. At this point, the most likely cause of route failure is the fact that a node on the route has failed or has moved out of range. An alternate route must be found "on the spot." The node thus broadcasts a message to its neighbors specifying that the next hop to a set of sources cannot be reached. Upon receiving this packet, each neighbor builds and unicasts the JOIN REPLY to its next hop if it has a route to the multicast sources. If no route is known, it simply broadcasts the packet specifying the next hop is not available. In both cases, the node sets its FG_FLAG. In practical implementations, this redundancy is sufficient to establish alternate paths until a more efficient route is established during the next refresh phase. The FG_FLAG setting of every neighbor may create excessive redundancy, but most of these settings will expire because only necessary forwarding group nodes will be refreshed in the next JOIN REPLY propagation phase.

2.4. Data Forwarding

After the group establishment and route construction process, a source can multicast packets to receivers via selected routes and forwarding groups. When receiving the multicast data packet, a node forwards it only when it is not a duplicate and the setting of the FG_FLAG for the multicast group has not expired. This procedure minimizes the traffic overhead and prevents sending packets through stale routes.

2.5. Soft State

In ODMRP, no explicit control packets need to be sent to join or leave the group. If a multicast source wants to leave the group, it simply stops sending JOIN QUERY packets since it does not have any multicast data to send to the group. If a receiver no longer wants to receive from a particular multicast group, it does not send the JOIN REPLY for that group. Nodes in the forwarding group are demoted to non-forwarding nodes if not refreshed (no JOIN REPLIES received) before they timeout.

2.6. Selection of Timer Values

Timer values for route refresh interval and forwarding group timeout interval can have impacts on ODMRP performance. The selection of these soft state timers should be adaptive to network environment (e.g., traffic type, traffic load, mobility pattern, mobility speed, channel capacity, etc.). When small route refresh interval values are used, fresh route and membership information can be obtained frequently at the expense of producing more packets and causing network congestion. On the other hand, when large route refresh values are selected, even though less control traffic will be generated, nodes may not know up-to-date route and multicast membership. Thus in highly mobile networks, using large route refresh interval values can yield poor protocol performance. The forwarding group timeout interval should also be carefully selected. In networks with heavy traffic load, small values should be used so that unnecessary nodes can timeout quickly and not create excessive redundancy. In situations with high mobility, however, large values should be chosen so that more alternative paths can be provided. It is important to note that the forwarding group timeout value must be larger (e.g., three to five times) than the value of route refresh interval.

2.7. Unicast Capability

One of the major strengths of ODMRP is its unicast routing capability [2,26]. Not only can ODMRP coexist with any unicast routing protocol, it can also operate very efficiently as an unicast routing protocol. Thus, a network equipped with ODMRP does not require a separate unicast protocol. Other ad hoc multicast routing protocols such as AMRoute [6], CAMP [14], RBM [8], and LAM [20] must be run on top of a unicast routing protocol. CAMP, RBM, and LAM in particular, only work with certain underlying unicast protocols.

2.8. Data Structures

Network hosts running ODMRP are required to maintain the following data structures.

2.8.1. Route Table

A route table is created on demand and is maintained by each node. An entry is inserted or updated when a non-duplicate JOIN QUERY is received. The node stores the destination (i.e., the source of the JOIN QUERY) and the next hop to the destination (i.e., the node which the JOIN QUERY is received from). The route table provides the next hop information when transmitting JOIN REPLIES.

2.8.2. Forwarding Group Table

When a node is a forwarding group node of a multicast group, it maintains the group information in the forwarding group table. The multicast group ID and the time when the node was last refreshed are recorded.

2.8.3. Message Cache

The message cache is maintained by each node to detect duplicates. When a node receives a new JOIN

QUERY or data packet, it stores the source ID and the sequence number of the packet. Note that entries in the message cache need not be maintained permanently. Schemes such as LRU (Least Recently Used) or FIFO (First In First Out) can be employed to expire and remove old entries in order to prevent the size of the message cache to be extensive.

3. Mobility Prediction

3.1. Adapting the Refresh Internal via Mobility Prediction

ODMRP requires periodic flooding of JOIN QUERY to refresh routes and group membership. Excessive flooding, however, is not desirable in ad hoc networks because of bandwidth constraints. Furthermore, flooding often causes congestion, contention, and collisions. Finding the optimal refresh interval is critical in ODMRP performance. Here we propose a scheme that adapts the refresh interval to mobility patterns and speeds. By utilizing the location and mobility information provided by GPS (Global Positioning System) [22], we predict the duration of time routes will remain valid.[1] With the predicted time of route disconnection, JOIN QUERIES are sent only when route breaks of ongoing data sessions are imminent.

In our prediction method, we assume a free space propagation model [36], where the received signal strength solely depends on its distance to the transmitter. We also assume that all nodes in the network have their clock synchronized (e.g., by using the NTP (Network Time Protocol) [30] or the GPS clock itself).[2] Therefore, if the motion parameters of two neighbors (e.g., speed, direction, radio propagation range, etc.) are known, we can determine the duration of time these two nodes will remain connected. Assume two nodes i and j are within the transmission range r of each other. Let (x_i, y_i) be the coordinate of mobile host i and (x_j, y_j) be that of mobile host j. Also let v_i and v_j be the speeds, and θ_i and θ_j $(0 \leq \theta_i, \theta_j < 2\pi)$ be the moving directions of nodes i and j, respectively. Then, the amount of time that they will stay connected, D_t, is predicted by:

$$D_t = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2 - (ad-bc)^2}}{a^2+c^2} \quad (3.1)$$

where

$a = v_i \cos\theta_i - v_j \cos\theta_j$,
$b = x_i - x_j$,
$c = v_i \sin\theta_i - v_j \sin\theta_j$, and
$d = y_i - y_j$.

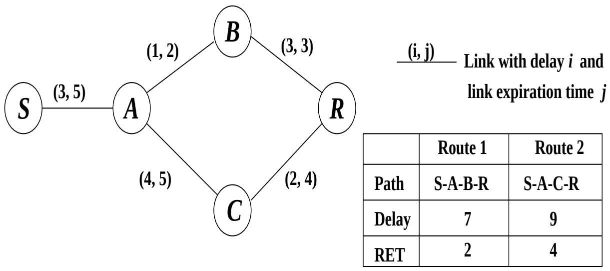
Note that when $v_i = v_j$ and $\theta_i = \theta_j$, D_t is set to ∞ without applying the above equation.

To utilize the information obtained from the prediction, extra fields must be added into JOIN QUERY and JOIN REPLY packets. When a source sends a JOIN QUERY, it appends its location, speed, and direction. It sets the MIN_LET (Minimum Link Expiration Time) field to the `MAX_LET_VALUE` since the source does not have any previous hop node. The next hop neighbor, upon receiving a JOIN QUERY, predicts the link expiration time between itself and the previous hop using the equation (3.1). The minimum between this value and the MIN_LET indicated by the JOIN QUERY is included in the packet. The rationale is that as soon as a single link on a path is disconnected, the entire path is invalidated. The node also overwrites the location and mobility information field written by the previous node with its own information. When a multicast member receives the JOIN QUERY, it calculates the predicted LET of the last link of the path. The minimum between the last link expiration time and the MIN_LET value specified in the JOIN QUERY is the RET (Route Expiration Time). This RET value is enclosed in the JOIN REPLY and broadcasted. If a forwarding group node receives multiple JOIN REPLIES with different RET values (i.e., lies in paths from the same source to multiple receivers), it selects the minimum RET among them and sends its own JOIN REPLY with the chosen RET value attached. When the source receives JOIN REPLIES, it selects the minimum RET among all the JOIN REPLIES received. Then the source can build new routes by flooding a JOIN QUERY before the minimum RET approaches (i.e., route breaks).

In addition to the estimated RET value, other factors need to be considered when choosing the refresh interval. If the node mobility rate is high and the topology changes frequently, routes will expire quickly

[1] Mobility speed and heading information can be obtained from GPS or the node's own instruments and sensors (e.g., campus, odometer, speed sensors, etc.).

[2] Time synchronization of the nodes is done only at the boot time. Once nodes have powered up and their clocks are synchronized, it is not required to perform periodic updates (although periodic updates can still be done in large intervals).



	Route 1	Route 2
Path	S-A-B-R	S-A-C-R
Delay	7	9
RET	2	4

Figure 6. Route selection example.

and often. The source may propagate Join Queries excessively and this excessive flooding can cause collisions and congestion, and clogs the network with control packets. Thus, the `MIN_REFRESH_INTERVAL` should be enforced to avoid control message overflow. On the other hand, if nodes are stationary or move slowly and link connectivity remains unchanged for a long duration of time, routes will hardly expire and the source will rarely send Join Queries. A few problems arise in this situation. First, if a node in the route suddenly changes its movement direction or speed, the predicted RET value becomes obsolete and routes will not be reconstructed in time. Second, when a non-member node which is located remotely to multicast members wants to join the group, it cannot inform the new membership or receive multicast data until a Join Query is received. Hence, the `MAX_REFRESH_INTERVAL` should be set. The selection of the `MIN_REFRESH_INTERVAL` and the `MAX_REFRESH_INTERVAL` values should be adaptive to network environments.

3.2. Route Selection Criteria

In the basic ODMRP, a multicast receiver selects routes based on the minimum delay (i.e., routes taken by the first Join Query received). A different route selection method can be applied when we use the mobility prediction. The idea is inspired by the Associativity-Based Routing (ABR) protocol [40] which chooses associatively stable routes. In our new algorithm, instead of using the minimum delay path, we choose a route that is the most stable (i.e., the one with the largest RET). To select a route, a multicast receiver must wait for an appropriate amount of time after receiving the first Join Query so that all possible routes and their RETs will be known. The receiver then chooses the most stable route and broadcasts a Join Reply. Route breaks will occur less often and the number of Join Query propagation will reduce because stable routes are used. An example showing the difference between two route selection algorithms is presented in Figure 6. Two routes are available from the source S to the receiver R. Route 1 has the path of $<S\text{-}A\text{-}B\text{-}R>$ and route 2 has the path of $<S\text{-}A\text{-}C\text{-}R>$. If the minimum delay is used as the route selection metric, the receiver node R selects route 1. Route 1 has a delay of 7 $(3+1+3=7)$ while route 2 has a delay of 9 $(3+4+2=9)$. Since the Join Query that takes route 1 reaches the receiver first, node R chooses route 1. If the stable route is selected instead, route 2 is chosen by the receiver. The route expiration time of route 1 is 2 $(\min(5,2,3)=2)$ while that of route 2 is 4 $(\min(5,5,4)=4)$. The receiver selects the route with the maximum RET, and hence route 2 is selected.

3.3. Alternative Method of Prediction

Since GPS may not work properly in certain situations (e.g., indoor, fading, etc.), we may not always be able to accurately predict the link expiration time for a particular link. However, there is an alternative method to predict the LET. This method is based on a more realistic propagation model and has been proposed in [1] and [33]. Basically, transmission power samples are measured periodically from packets received from a node's neighbor. From this information it is possible to compute the rate of change for a particular neighbor's transmission power level. Therefore, the time when the transmission power level will drop below the acceptable value (i.e., hysteresis region) can be predicted. We plan to investigate this option in our future work.

4. Simulation Model and Methodology

The simulator for evaluating ODMRP and other multicast routing protocols was implemented within the GloMoSim library [41]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC [3]. Our simulation modeled a network of 50 mobile hosts placed randomly within a $1000m \times 1000m$ area. Radio propagation range for each node was 250 meters and channel capacity was 2 Mbps. There were no network partitions throughout the simulation and the average number of neighbors for each node was 6.82. Each simulation executed for 600 seconds of simulation time. Multiple runs with different seed numbers were conducted for each scenario and collected data were averaged over those runs.

4.1. Channel and Radio Model

We used a free space propagation model [36] with a threshold cutoff in our experiments. In the free space model, the power of a signal attenuates as $1/d^2$ where d is the distance between radios. In addition to the free space channel model, we also implemented SIRCIM (Simulation of Indoor Radio Channel Impulse-response Models) [37] which considers multipath fading, shadowing, barriers, foliages, etc. SIRCIM is more accurate than the free space model, but we decided against using SIRCIM in our study because: (a) the complexity of SIRCIM increases simulation time by three orders of magnitude; (b) the accuracy of the channel model does not affect the relative ranking of the multicast protocols evaluated in this study; and (c) SIRCIM must be "tuned" to the characteristics of the physical environment (e.g., furniture, partitions, etc.), thus requiring a much more specific scenario than we are assuming in our experiments.

In the radio model, we assumed the ability of a radio to lock onto a sufficiently strong signal in the presence of interfering signals, i.e., radio capture. If the capture ratio (the ratio of an arriving packet's signal strength over the sum of all colliding packets) [36] was greater than a predefined threshold value, the packet was received while all other interfering packets were dropped.

4.2. Medium Access Control Protocol

The IEEE 802.11 MAC with Distributed Coordination Function (DCF) [18] was used as the MAC protocol. DCF is the mode which allows nodes to share the wireless channel in an ad hoc configuration. The specific access scheme is Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with acknowledgments. Optionally, the nodes can make use of Request To Send/Clear To Send (RTS/CTS) channel reservation control frames for unicast, virtual carrier sense, and fragmentation of packets larger than a given threshold. By setting timers based upon the reservations in RTS/CTS packets, the virtual carrier sense augments the physical carrier sense in determining when mobile nodes perceive that the medium is busy. Fragmentation is useful in the presence of high bit error and loss rates, as it reduces the size of the data units that need to be retransmitted.

In our experiments, we employed RTS/CTS exclusively for unicast control packets directed to specific neighbors (e.g., replies). All other transmissions use CSMA/CA. We chose this configuration to minimize the frequency and deleterious effects of collisions over the wireless medium. We did not employ fragmentation because our data packets were small enough that the additional overhead would reduce overall network throughput.

4.3. Multicast Protocols

In addition to ODMRP, we implemented four multicast protocols for ad hoc networks: AMRoute [6], AMRIS [42], CAMP [14], and flooding. When implementing the protocols, we followed the specifications of each protocol as defined in the published literature. We directly queried the protocol designers about details which were not specified in the publications (e.g., various timer values, core selection algorithm, etc.). ODMRP and AMRIS do not require underlying unicast protocol to operate, but AMRoute and CAMP do. While AMRoute can work with any protocol, the designers of CAMP specifically state that it can operate only with certain unicast protocols [14]. We have implemented one of those protocols, WRP [32], a distance-vector based unicast routing protocol developed by the same group which developed CAMP. For a fair comparison, WRP was used as the underlying unicast protocol also for AMRoute. For the ODMRP [28] implementation, we used the version *without* mobility prediction (i.e., network hosts are *not* equipped with GPS).[3] The parameter values used for each protocol in our simulation are shown from Tables 1 to 4.

Table 5 summarizes key characteristics and properties of the protocols we simulated.[4] Note that ODMRP requires periodic messaging of JOIN QUERY only when sources have data packets to send.

[3] We have decided not to utilize the mobility prediction in ODMRP implementation to give fair comparisons to other protocols. Performance improvements made by the mobility prediction of ODMRP has been reported in [27].

[4] For the detail operations of each protocol, readers are referred to published documents of each protocol.

Table 1
Parameter values for AMRoute.

JOIN-REQ interval	60 sec
JOIN-REQ interval when no group members are connected to the core	5 sec
TREE-CREATE interval	20 sec
TREE-CREATE timeout	40 sec
Core resolution algorithm	Highest ID

Table 2
Parameter values for ODMRP.

JOIN QUERY refresh interval	3 sec
JOIN REPLY acknowledgment timeout	25 ms
Maximum JOIN REPLY retransmissions	3

Table 3
Parameter values for AMRIS.

Beacon interval	1 sec
Max allowed beacon losses	3
NEW SESSION lifetime	3 sec
JOIN-REQ acknowledgment timeout	2 sec
Broadcast random jitter time	50 ms

Table 4
Parameter values for CAMP.

Number of cores	1
Beacon interval	3 sec
Update interval	3 sec
Age out anchor timeout	45 sec
Heartbeat interval	15 sec
Request retransmission interval	9 sec
Max JOIN REQUEST retransmissions	3

4.4. Traffic Pattern

A traffic generator was developed to simulate constant bit rate sources. The size of data payload was 512 bytes. The senders were chosen randomly among multicast members who in turn were chosen with uniform probability among 50 network hosts. The member nodes join the multicast session at the beginning of the simulation and remain as members throughout the simulation. Hence, the simulation experiments do not test/account for the overhead produced in the session leave process.

Table 5
Summary of protocols.

Protocols	AMRoute	ODMRP	AMRIS	CAMP	Flood
Configuration	Tree	Mesh	Tree	Mesh	Mesh
Loop-Free	No	Yes	Yes	Yes	Yes
Dependency on Unicast Protocol	Yes	No	No	Yes	No
Periodic Messaging	Yes	Yes	Yes	Yes	No
Control Packet Flood	Yes	Yes	Yes	No	No

4.5. Metrics

We have used the following metrics in comparing protocol performance. Some of these metrics were suggested by the IETF MANET working group for routing/multicasting protocol evaluation [10].

- **Packet delivery ratio**: The ratio of the number of data packets delivered to the destinations versus the number of data packets supposed to be received. This number presents the effectiveness of a protocol.
- **Number of data packets transmitted per data packet delivered**: Data packets transmitted is the count of every individual transmission of data by each node over the entire network. This count includes transmissions of packets that are eventually dropped and retransmitted by intermediate nodes. Note that in unicast protocols, this measure is always equal or greater than one. In multicast, since a single transmission can deliver data to multiple destinations, the measure can be less than one.
- **Number of control bytes transmitted per data byte delivered**: Instead of using a measure of pure control overhead, we chose to use the ratio of control bytes transmitted to data byte delivered to investigate how efficiently control packets are utilized in delivering data. Note that not only bytes of control packets (e.g., beacons, route updates, join requests, acknowledgments, etc.), but also bytes of data packet

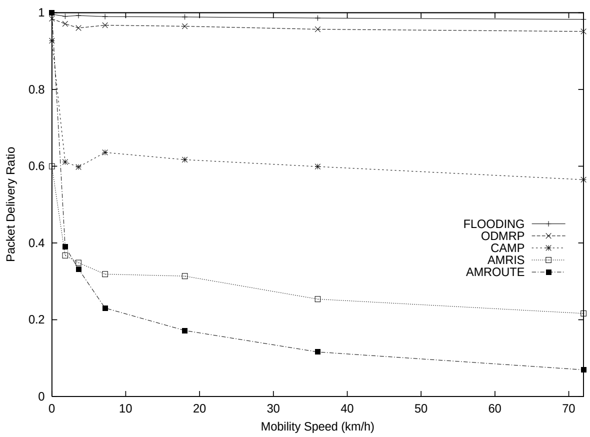


Figure 7. Packet delivery ratio as a function of mobility speed.

headers are included in the number of control bytes transmitted. Accordingly, only the data payload bytes contribute to the data bytes delivered.

- **Number of control and data packets transmitted per data packet delivered**: This measure shows the efficiency in terms of channel access and is very important in ad hoc networks since link layer protocols are typically contention-based.

5. Simulation Results

We tried to emulate as many scenarios as possible to investigate the protocol performance under different network situations. We have varied the following four items: mobility speed, number of multicast senders, multicast group size, and network traffic load.

5.1. Mobility Speed

5.1.1. Scenarios

Each node moved constantly with the predefined speed. Moving directions of each node were selected randomly, and when nodes reached the simulation terrain boundary, they bounced back and continued to move. The node movement speed was varied from 0 km/hr to 72 km/hr. In the mobility experiment, twenty nodes are multicast members and five sources transmit packets at the rate of 2 pkt/sec each.

5.1.2. Results and Analysis

Figure 7 illustrates the packet delivery ratio of the protocols under different speeds. ODMRP shows good performance even in highly dynamic situations. ODMRP provides redundant routes with a mesh topology and the chances of packet delivery to destinations remain high even when the primary routes are unavailable. The path redundancy enables ODMRP to suffer only minimal data loss and be robust to mobility. In fact, ODMRP was as effective as flooding in this experiment.

CAMP, which also uses a mesh topology, shows a better performance than protocols that use trees. However, CAMP exhibited poorer performance than we had expected, especially under mobility. A major reason CAMP was not as effective as ODMRP was that many packets headed to distant routers in the mesh were not delivered. In CAMP, since the paths to distant destinations have fewer redundant paths than those closer to the center of the mesh, they are more prone to occasional link breaks preventing a vital “anchoring” node from successfully receiving packets. Most of the successful packet transmissions occur in this mesh center, and require fewer data transmissions per delivery than transmissions to the mesh edges. In addition, in the presence of mobility and link breaks, WRP, which is the unicast protocol CAMP prefers to coexist with, can require a period of network re-convergence in regards to a subset of destinations. During this interval, this subset of destinations will be marked as unreachable by the loop-detection facilities. If the group core is a part of this subset of temporarily unreachable nodes, the multicast routing updates regarding mesh maintenance will be postponed, which also contributes to delays in mesh response to mobility.

AMRIS shows a poor delivery ratio compared with protocols that use mesh configuration. Since AMRIS builds a shared tree for data dissemination, there is only one path between member nodes. If a single tree link breaks because of node movements, packet collision, or congestion, destinations can not receive packets. AMRIS detects node movements and tree breaks by a beaconing mechanism. Nodes send beacons every second, and neighbors are considered to have moved away if three consecutive beacons are not received. Thus, in the best case, it takes three seconds after the link break for AMRIS to start tree readjustment. A number of packets can be lost during that period. There are possible solutions to this problem, but they all have respective drawbacks. If beacons are sent more often, that could increase packet collisions. If the number of allowed beacon losses is decremented, a node may attempt to find a new route when the link is not broken but beacons are lost due to collisions. Finding the optimal beacon interval and allowed number of beacon losses for AMRIS is beyond the scope of the paper and we used the values recommended by the AMRIS designers. The result that surprised us was for zero mobility. While other protocols showed data delivery ratio approaching unity, AMRIS delivered only 60% of data packets. Since each node sends beacons every second, there are a number of packets contending for the channel. The beacon size of AMRIS is relatively large compared with other protocols that send beacons (see [42]). Thus, the beacon traffic combined with the data traffic causes a large number of collisions leading to 40% drop. Under very light data traffic, AMRIS shows improved performance as will be shown in Figure 14.

AMRoute was the least effective of the protocols with mobility. Although its delivery ratio is near perfect in no mobility, it fails to deliver a significant number of packets even at low mobility speeds. The delivery ratio steadily worsens as the mobility speed is increased. One of the reasons AMRoute performs so poorly is due to the formation of loops and the creation of sub-optimal trees when mobility is present (at 72 km/hr, the average hop count was nearly eight while other protocols were below four). Loops occur during the tree reconstruction phase when some nodes are forwarding data according to the stale tree and others according to the newly built tree. The existence of loops is critical in protocol performance because they cause serious congestion. At some instants, nodes had up to 13.75 packets dropped per second. The loss of packets due to buffer overflow has two consequences. First, if a data packet is dropped in the early stage of its multicast tree traversal, a large portion of tree members will not receive it. Second, if control packets (TREE-CREATE, JOIN-ACK, etc.) are dropped, the tree is not properly built or becomes segmented and data will not be delivered. Another reason for AMRoute ineffectiveness is its dependency on the underlying unicast protocol. AMRoute relies on the unicast protocol to set up bidirectional tunnels between group members for the multicast tree. However, as shown in [35], when mobility speed increases, the bidirectional link assumption in ad hoc networks becomes weak (i.e., a node can reach a neighboring node, but not necessarily vice versa). In our experiments, unidirectional "critical" links existed in AMRoute trees.

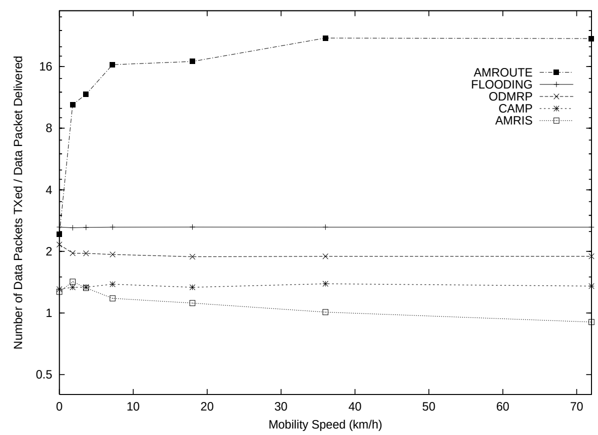


Figure 8. Number of data packets transmitted per data packet delivered as a function of mobility speed.

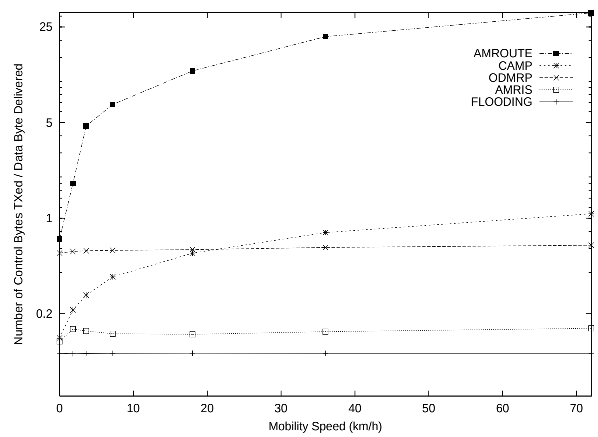


Figure 9. Number of control bytes transmitted per data byte delivered as a function of mobility speed.

Critical links are such that packets sent by the one end of the link are mostly received by the other end but not vice versa. A great number of packets are lost at these critical links. Since there are no alternate routes in the AMRoute shared tree (although AMRoute creates the mesh in order to build a tree, data is forwarded only by tree nodes), data delivery ratio is very low.

Figure 8 shows the number of data transmissions per data delivery to destinations. AMRoute has the highest number of transmissions because of loops. We can observe that protocols using meshes (i.e., ODMRP and CAMP) transmit more data packets than AMRIS, which uses a tree. In fact, ODMRP transmits nearly as much data as flooding because it exploits multiple redundant routes for data delivery.

The control byte overhead per data byte delivered is shown in Figure 9. Remember that data packet header is included in control overhead. Flooding has no con-

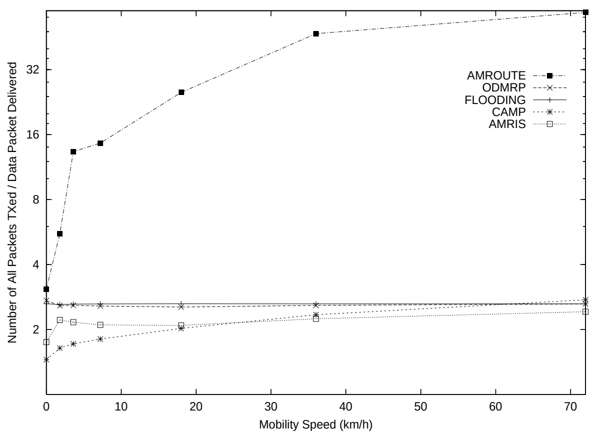


Figure 10. Number of total packets transmitted per data packet delivered as a function of mobility speed.

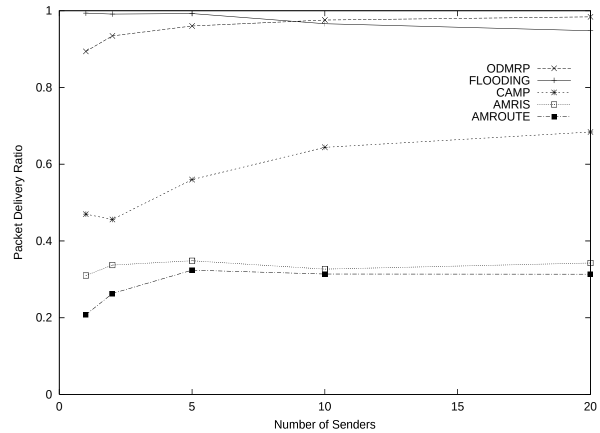


Figure 11. Packet delivery ratio as a function of number of senders.

trol packets. Hence, only the data header contributes to control overhead and this overhead does not increase with mobility. Other protocols generate increasing overhead as speed increases. AMRIS shows a low control overhead compared with other multicast schemes. The primary reason is that it transmitted less data packets (as seen in Figure 8). CAMP shows a larger control overhead under high mobility than ODMRP because of its reliance on the unicast routing protocol WRP, which sends triggered updates. WRP suffers from exponential growth in control traffic overhead under increasing mobility. Moreover, CAMP piggybacks its own update messages onto WRP updates and those packets play a role in overhead growth. In ODMRP, the control overhead remains relatively constant because no updates are triggered by mobility. JOIN QUERY refresh interval was set constant to three seconds and hence no additional overhead is required as mobility increases. AMRoute has the highest ratio because of the data headers that are caught in the loops. The high ratio is also due to the formation of inefficient trees. During the tree creation phase, an inefficient tree can be formed when the TREE-CREATE packets from distant mesh neighbors arrives earlier than packets from nearby nodes (e.g., due to network congestion, etc.). The non-optimal tree results in having longer hops between member nodes and increasing the number of data transmissions.

The number of all packets transmitted per data packet delivered is presented in Figure 10. An interesting result is that CAMP has a smaller number of transmissions than ODMRP. This result stems from two factors. First, ODMRP transmits more data packets on redundant paths than CAMP. Second, although CAMP has more control overhead bytes, the number of control packet transmissions is lower since CAMP updates are piggybacked onto WRP updates. Again, AMRIS has the smallest number of packet transmissions because it uses a tree and AMRoute has the highest value because of loops.

5.2. Number of Senders

5.2.1. Scenarios

In this experiment, the multicast group size is set constant at twenty, node mobility speed is slow (1 m/s), and network traffic load is relatively light (10 pkt/sec). The number of multicast senders range in the set {1, 2, 5, 10, 20}. A single sender represents a class lecture scenario, while at the other extreme, twenty senders model a video conference situation.

5.2.2. Results and Analysis

The packet delivery ratio as a function of the number of multicast senders is shown in Figure 11. As the number of sources increases, performance of flooding slightly degrades as more packets are lost by collision, congestion, and channel contention. ODMRP shows robustness to the number of sources. In fact, performance even improves with number of senders because of increasing number of forwarding nodes and thus better path redundancy. ODMRP limits the number of sources that can send JOIN QUERIES at the same time. Whenever a source needs to flood a JOIN QUERY, it listens if any other source is flooding the packet. It proceeds to send the JOIN QUERY only if no flooded packets are received within a certain period. Thus, the number of collisions

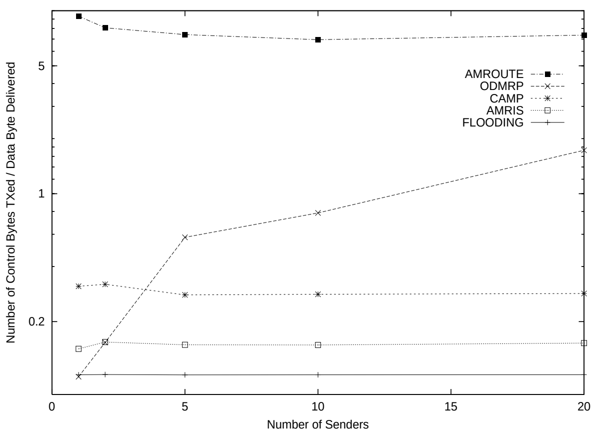


Figure 12. Number of control bytes transmitted per data byte delivered as a function of number of senders.

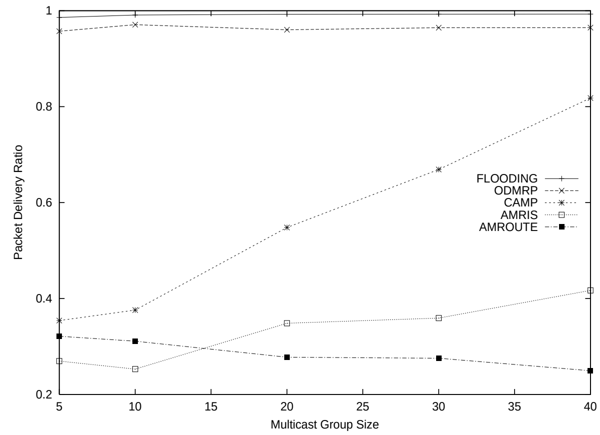


Figure 13. Packet delivery ratio as a function of multicast group size.

decreases and the the protocol remains effective. Like ODMRP, CAMP shows improved performance with a larger number of senders due to the increase in the number of anchors that each node requires. Each member node requests every neighbor which is in the reverse shortest path to some source, to rebroadcast multicast update packets it receives initially. Hence increasing the number of sources increases the redundant paths in the mesh. AMRIS and AMRoute performance was unaffected by the number of senders because they use a shared tree for the multicast session.

Figure 12 shows the control overhead per data byte delivered. Every protocol except ODMRP shows a relatively constant value. While the other three multicast protocols form a shared mesh or tree, ODMRP builds per-source meshes. If the number of senders increases, more JOIN QUERY packets are propagated and control overhead grows accordingly. We can speculate from this result that ODMRP in its present form may not be as efficient in networks where a large number of nodes (e.g., hundreds and thousands) are multicast sources.

5.3. Multicast Group Size

5.3.1. Scenarios

We varied the number of multicast members to investigate the scalability of the protocol. While fixing the number of senders at five, mobility speed at 1 m/s, and network traffic rate at 10 pkt/sec, the multicast group size was varied from five to forty members.

5.3.2. Results and Analysis

The routing effectiveness of protocols as a function of multicast group size is illustrated in Figure 13. Flooding and ODMRP performance were not affected by the number of multicast members. CAMP, on the other hand, performs markedly better as the number of receivers increases. Since the mesh becomes massive with the growth of the members, more redundant routes are formed and that improves the performance. If only a small number of nodes join the multicast session, the mesh actually appears closer to a tree for distant nodes, and the performance is reflected in this graph. AMRIS also shows improvements with the member size growth, but they are less dramatic than CAMP because redundant routes are not established in AMRIS. AMRoute shows the complete opposite behavior. As the group size increases, the delivery ratio actually drops. This behavior is due to the "critical" links that exist in the AMRoute multicast tree (critical links were described in Section 5.1). As the group size increases, the number of tree links increases and the probability of sources being isolated in the tree by critical links increases as well.

5.4. Network Traffic Load

5.4.1. Scenarios

To study the impact of data traffic load on multicast protocols, we varied the load on the network. There were five senders and the multicast group size was twenty. In this experiment, there was no node mobility. Therefore, the packet drops are only caused by buffer overflow, collision, and congestion. The network traffic

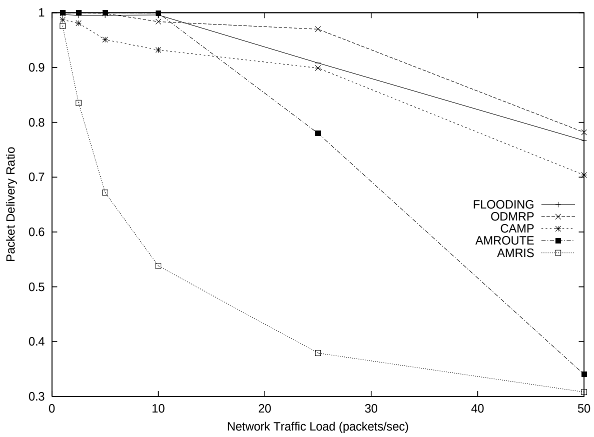


Figure 14. Packet delivery ratio with no mobility as a function of network traffic load.

loads used were between 1 pkt/sec and 50 pkt/sec.

5.4.2. Results and Analysis

Packet delivery ratios for various traffic loads are shown in Figure 14. AMRIS was the most sensitive to traffic load. AMRIS delivers a high percentage of data packets in extremely light load (i.e., less than 5 pkt/sec). As the load increases however, the ratio drops rapidly. As explained in Section 5.1, the transmission and the size of beacons resulted in numerous packet collisions. AMRoute performance is nearly perfect when the packet rate is relatively low, but it drops rather quickly when the traffic load is increased. The degradation is caused by buffer overflow at the members in the tree and at the mesh nodes that connect the tree members. CAMP performance is also affected by traffic load. As the load increases, the number of collisions and packet losses increase. When important control packets are dropped, anchor construction can be delayed and data packets can fail to reach all the anchors. The degradation follows a pattern similar to flooding and ODMRP, indicating a common behavior in mesh based data delivery. Flooding shows worse delivery ratios than ODMRP as load grows. Since every data packet is flooded, the number of collisions and buffer overflows grows with the load. ODMRP is also affected by load, but the packet loss rate is less severe than flooding because the number of data packet transmissions is less than flooding. Although ODMRP shows the same patters of behaviors as CAMP, it gives a better delivery rate because it has less control overhead and suffers less buffer overflows than CAMP.

6. Conclusions

We have presented ODMRP for multihop wireless mobile networks. ODMRP builds and maintains a mesh for each multicast group. Providing multiple paths by the formation of mesh configuration makes the protocol robust to mobility. Alternate routes enable data delivery in the face of mobility and link breaks while the primary route is being reconstructed. The protocol does not yield excessive channel overhead in highly mobile networks because no control packets are triggered by link breaks. ODMRP also applies demand-driven, as opposed to periodic, multicast route construction and takes soft state approach in membership maintenance. The key properties of ODMRP are:

- Simplicity
- Low channel and storage overhead
- Usage of up-to-date shortest routes
- Reliable construction of routes and forwarding group
- Robustness to host mobility
- Maintenance and utilization of multiple paths
- Exploitation of the broadcast nature of the wireless environment
- Unicast routing capability

Simulation results indicate that mesh based protocols outperform tree based protocols significantly. In addition, compared with another mesh protocol CAMP, ODMRP produced less control overhead and efficiently utilized those control packets to deliver more data packets to multicast members. Since the primary concerns of ad hoc networks are frequent topology changes and constrained bandwidth, it is critical that the protocol supplies multiple routes and yields minimal overhead. ODMRP therefore, is an attractive choice for multicasting in ad hoc wireless networks. The protocol, however, may suffer from excessive flooding when there are a large number of multicast senders. We are currently developing enhancements to make ODMRP more scalable to large member groups.

References

[1] P. Agrawal, D.K. Anvekar, and B. Narendran, "Optimal Prioritization of Handovers in Mobile Cellular Networks," *Proceedings of IEEE PIMRC'94*, The Hague, Netherlands, Sep. 1994, pp. 1393-1398.

[2] S.H. Bae, S.-J. Lee, and M. Gerla, "Unicast Performance Analysis of the ODMRP in a Mobile Ad hoc Network Testbed," *Proceedings of IEEE ICCCN 2000*, Las Vegas, NV, Oct. 2000, pp. 148-153.

[3] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song, "PARSEC: A Parallel Simulation Environment for Complex Systems," *IEEE Computer*, vol. 31, no. 10, Oct. 1998, pp.77-85.

[4] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT) - An Architecture for Scalable Inter-Domain Multicast Routing," *Proceedings of ACM SIGCOMM'93*, San Francisco, CA, Oct. 1993, pp. 85-95.

[5] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," *Proceedings of ACM SIGCOMM'94*, London, UK, Sep. 1994, pp. 212-225.

[6] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Adhoc Multicast Routing Protocol," *Internet-Draft*, `draft-talpade-manet-amroute-00.txt`, Aug. 1998, Work in progress.

[7] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks," *Baltzer Cluster Computing*, vol. 1, no. 2, 1998, pp. 187-196.

[8] M.S. Corson and S.G. Batsell, "A Reservation-Based Multicast (RBM) Routing Protocol for Mobile Networks: Initial Route Construction Phase," *ACM/Baltzer Wireless Networks*, vol. 1, no. 4, Dec. 1995, pp. 427-450.

[9] M.S. Corson and A. Ephremides, "A Distributed Routing Algorithm for Mobile Wireless Networks," *ACM/Baltzer Wireless Networks*, vol. 1, no. 1, Feb. 1995, pp.61-81.

[10] M.S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *Request For Comments 2501*, Internet Engineering Task Force, Jan. 1999.

[11] S.E. Deering and D.R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," *ACM Transactions on Computer Systems*, vol. 8, no. 2, May 1990, pp. 85-110.

[12] S. Deering, D.L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, "The PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, Apr. 1996, pp. 153-162.

[13] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, Apr. 1997, pp. 277-290.

[14] J.J. Garcia-Luna-Aceves and E.L. Madruga, "The Core-Assisted Mesh Protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, Aug. 1999, pp. 1380-1394.

[15] J.J. Garcia-Luna-Aceves and M. Spohn, "Source-Tree Routing in Wireless Networks," *Proceedings of IEEE ICNP'99*, Toronto, Canada, Nov. 1999, pp. 273-282.

[16] J.J. Garcia-Luna-Aceves and M. Spohn, "Scalable Link-State Internet Routing," *Proceedings of IEEE ICNP'98*, Austin, TX, Oct. 1998, pp. 52-61.

[17] M. Gerla, C.-C. Chiang, and L. Zhang, "Tree Multicast Strategies in Mobile, Multihop Wireless Networks," *ACM/Baltzer Mobile Networks and Applications*, vol. 4, no. 3, Oct. 1999, pp. 193-207.

[18] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification*, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.

[19] Internet Engineering Task Force (IETF) Mobile Ad Hoc Networks (MANET) Working Group Charter, `http://www.ietf.org/html.charters/manet-charter.html`.

[20] L. Ji and M.S. Corson, "A Lightweight Adaptive Multicast Algorithm," *Proceedings of IEEE GLOBECOM'98*, Sydney, Australia, Nov. 1998, pp. 1036-1042.

[21] J. Jubin and J.D. Tornow, "The DARPA Packet Radio Network Protocols," *Proceedings of the IEEE*, vol. 75, no. 1, Jan. 1987, pp. 21-32.

[22] E.D. Kaplan (Editor), *Understanding the GPS: Principles and Applications*, Artech House, Boston, MA, Feb. 1996.

[23] L. Kleinrock and J. Silvester, "Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number," *Proceedings of National Telecommunications Conference*, Birmingham, AL, Dec. 1978, pp. 4.3.2-4.3.5.

[24] L. Kleinrock and F.A. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," *IEEE Transactions on Communications*, vol. COM-23, no. 12, Dec. 1975, pp. 1400-1416.

[25] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol," *Proceedings of IEEE WCNC'99*, New Orleans, LA, Sep. 1999, pp. 1298-1302.

[26] S.-J. Lee, W. Su, and M. Gerla, "Exploiting the Unicast Functionality of the On-Demand Multicast Routing Protocol," *Proceedings of IEEE WCNC 2000*, Chicago, IL, Sep. 2000.

[27] S.-J. Lee, W. Su, and M. Gerla, "Ad hoc Wireless Multicast with Mobility Prediction," *Proceedings of IEEE ICCCN'99*, Boston, MA, Oct. 1999, pp. 4-9.

[28] S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *Internet Draft*, `draft-ietf-manet-odmrp-02.txt`, Jan. 2000, Work in progress.

[29] D.A. Maltz, J. Broch, J. Jetcheva, and D.B. Johnson, "The Effects of On-Demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, Aug. 1999, pp. 1439-1453.

[30] D.L. Mills, "Internet Time Synchronization: the Network Time Protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, Oct. 1991, pp. 1482-1493.

[31] J. Moy, "Multicast Routing Extensions for OSPF," *Communications of the ACM*, vol. 37, no. 8, Aug. 1994, pp. 61-66, 114.

[32] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM/Baltzer Mobile*

Networks and Applications, vol. 1, no. 2, Oct. 1996, pp. 183-197.

[33] B. Narendran, P. Agrawal, and D.K. Anvekar, "Minimizing Cellular Handover Failures Without Channel Utilization Loss," *Proceedings of IEEE GLOBECOM'94*, San Francisco, CA, Dec. 1994, pp. 1679-1685.

[34] T. Ozaki, J.B. Kim, and T. Suda, "Bandwidth-Efficient Multicast Routing Protocol for Ad-Hoc Networks," *Proceedings of IEEE ICCCN'99*, Boston, MA, Oct. 1999, pp. 10-17.

[35] R. Prakash, "Unidirectional Links Prove Costly in Wireless Ad-Hoc Networks," *Proceedings of ACM DIAL M'99 Workshop*, Seattle, WA, Aug. 1999, pp. 15-22.

[36] T.S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, Oct. 1995.

[37] T.S. Rappaport, S.Y. Seidel, and K. Takamizawa, "Statistical Channel Impulse Response Models for Factory and Open Plan Building Radio Communication System Design," *IEEE Transactions on Communications*, vol. COM-39, no. 5, May 1991, pp. 794-807.

[38] E.M. Royer and C.E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," *Proceedings of ACM/IEEE MOBICOM'99*, Seattle, WA, Aug. 1999, pp. 207-218.

[39] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing," *Proceedings of IEEE WCNC'99*, New Orleans, LA, Sep. 1999, pp. 1313-1317.

[40] C.-K. Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks," *Wireless Personal Communications Journal*, Kluwer Academic Publishers, vol. 4, no. 2, Mar. 1997, pp. 103-139.

[41] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, *GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems*, http://pcl.cs.ucla.edu/projects/domains/glomosim.html.

[42] C.W. Wu, Y.C. Tay, and C.-K. Toh, "Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional Specification," *Internet-Draft*, `draft-ietf-manet-amris-spec-00.txt`, Nov. 1998, Work in progress.